

Netra™ 240 Server Systemadministrationshandbuch

Sun Microsystems, Inc.
www.sun.com

Teilenummer: 817-5011-11
Juli 2004, Ausgabe A

Bitte senden Sie Ihre Anmerkungen zu diesem Dokument an: `http://www.sun.com/hwdocs/feedback`


Adobe PostScript

Inhalt

Abbildungen

Tabellen

Vorwort

Das *Netra 240 Server Systemadministrationshandbuch* wurde für erfahrene Systemverwalter entwickelt. Es umfasst eine allgemeine Beschreibung der Diagnosewerkzeuge des Netra™ 240 Servers und verschiedene Serveradministrationsaufgaben.

Die praktische Umsetzung der Informationen in dem vorliegenden Handbuch erfordert die Kenntnis von Computernetzwerkkonzepten und der zugehörigen Begriffe. Des Weiteren sollten Sie mit dem Betriebssystem Solaris™ (Solaris OS) eingehend vertraut sein.

Einige einleitende Worte

In diesem Handbuch wird weder die Serverinstallation noch die Rack-Montage behandelt. Weitere Einzelheiten zu diesen Themen finden Sie im *Netra 240 Server Installationshandbuch* (Teilenummer 817-5001-11).

Bevor Sie in diesem Buch beschriebene Maßnahmen durchführen, stellen Sie sicher, dass Sie zuerst die *Important Safety Information for Sun Hardware Systems* (Teilenummer 816-7190-10) *lesen.*

Verwenden von UNIX-Befehlen

Verwenden Sie diesen Abschnitt, um Leser darauf hinzuweisen, dass nicht alle UNIX-Befehle angegeben werden. Beispiel:

Dieses Dokument enthält unter Umständen keine Informationen zu grundlegenden UNIX®-Befehlen und -Verfahren (z. B. das Herunterfahren oder Starten des Systems und das Konfigurieren von Geräten). Weitere Informationen finden Sie hier:

- Software-Dokumentation, die Sie mit Ihrem System erhalten haben
- Solaris™ Betriebssystemdokumentation unter der folgenden Adresse:

  `http://docs.sun.com`

Shell-Eingabeaufforderungen

Shell	Eingabeaufforderung
C-Shell	*Computername*%
C-Shell-Superuser	*Computername*#
Bourne-Shell und Korn-Shell	$
Bourne-Shell- und Korn-Shell-Superuser	#

Typografische Konventionen

Schriftart[i]	Bedeutung	Beispiele
`AaBbCc123`	Namen von Befehlen, Dateien und Verzeichnissen in Bildschirmausgaben	Bearbeiten Sie die `.login`-Datei. Mit `ls -a` können Sie alle Dateien auflisten. `% Sie haben Post.`
`AaBbCc123`	Tastatureingaben im Gegensatz zu Bildschirmausgaben des Computers	`%` **`su`** `Passwort:`
AaBbCc123	Buchtitel, neu eingeführte Begriffe oder Betonungen. Ersetzen Sie die Befehlszeilenvariablen durch echte Namen oder Werte.	Siehe Kapitel 6 im *Benutzerhandbuch*. Diese Optionen werden als *Klassenoptionen* bezeichnet. Dazu *müssen* Sie als Superuser angemeldet sein. Geben Sie zum Löschen einer Datei `rm` *Dateiname* ein.

i Ihr Browser verwendet möglicherweise andere Einstellungen.

Zugehörige Dokumentation

Beschreibung	Titel	Teilenummer
Installation – Überblick	*Netra 240 Server Quick Start Guide*	817-3904-xx
Neueste Produktaktualisierungen	*Netra 240 Server-Versionshinweise*	817-3142-xx
Einhaltung technischer Vorschriften und Sicherheit	*Important Safety Information for Sun Hardware Systems*	816-7190-10
	Netra 240 Server Sicherheits- und Compliance-Handbuch	817-5018-11
Dokumentation, Website	*Sun Netra 240 Server Dokumentation*	817-2697-10
Installation	*Netra 240 Server Installationshandbuch*	817-5001-11
Verwaltung bei ausgefallener Stromversorgung	*Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch*	817-5003-11
Wartung und Reparatur	*Netra 240 Server Service Manual*	817-2699-xx

Zugriff auf die Sun-Dokumentation

Unter der folgenden Internet-Adresse können Sie eine breite Auswahl von Sun-Dokumentationen, einschließlich übersetzter Versionen, lesen, drucken oder kaufen:

`http://www.sun.com/documentation`

Websites anderer Hersteller

Sun ist nicht verantwortlich für die Verfügbarkeit der in diesem Dokument erwähnten Websites anderer Hersteller. Sun haftet nicht für den Inhalt oder Werbung auf diesen Websites oder für die auf diesen Websites angebotenen Produkte und Materialien. Sun übernimmt keine Verantwortung oder Haftung für tatsächliche oder angebliche Schäden oder Verluste, die im Zusammenhang mit den auf diesen Websites angebotenen Informationen, Waren oder Dienstleistungen entstanden sind.

Kontaktieren der technischen Unterstützung von Sun

Bei technischen Fragen zu diesem Produkt, die in diesem Dokument nicht beantwortet werden, finden Sie weitere Informationen unter:

`http://www.sun.com/service/contacting`

Kommentare und Anregungen

Wir bemühen uns um eine stetige Verbesserung unserer Dokumentation und freuen uns über Ihre Kommentare und Anregungen. Senden Sie uns Ihre Kommentare unter:

`http://www.sun.com/hwdocs/feedback`

Bitte geben Sie dabei den Titel und die Teilenummer Ihres Dokuments an:

Netra 240 Server Systemadministrationshandbuch, Teilenummer 817-5011-11

KAPITEL 1

Fehlerbehebungswerkzeuge

In diesem Kapitel werden die Diagnosewerkzeuge vorgestellt, die für den Netra 240 Server verfügbar sind. Es enthält die folgenden Abschnitte:

Überblick über die Diagnosewerkzeuge

Sun stellt eine Reihe von Diagnosewerkzeugen für den Netra 240 Server zur Verfügung, die in der folgenden Tabelle zusammengefasst sind.

TABELLE 1-1 Überblick über die Fehlerbehebungswerkzeuge

Diagnosewerkzeug	Typ	Beschreibung	Zugriff und Verfügbarkeit	Remote-Option
ALOM	Hardware und Software	Überwachung der Umgebungsbedingungen, grundlegende Fehlerisolierung, Zugriff von Remote-Konsole.	Funktionsfähig bei Standby-Versorgung und ohne Betriebssystem.	Für Remote-Access konzipiert.
LEDs	Hardware	Statusanzeige des Gesamtsystems und bestimmter Komponenten.	Zugriff über Systemgehäuse. Verfügbar bei Stromversorgung.	Lokal, kann aber mithilfe von ALOM angezeigt werden.
Selbsttest beim Einschalten (Power-On Self-Test, POST)	Firmware	Test der Hauptsystemkomponenten.	Startet automatisch oder beim Systemstart. Verfügbar bei heruntergefahrenem Betriebssystem.	Lokal, kann aber mithilfe von ALOM angezeigt werden.
OpenBoot-Befehle	Firmware	Anzeige verschiedener Systeminformationen.	Verfügbar bei heruntergefahrenem Betriebssystem.	Lokal, kann aber mithilfe von ALOM geöffnet werden.
OpenBoot-Diagnoseroutinen (OpenBoot Diagnostics)	Firmware	Test von Systemkomponenten, insbesondere von Peripherie- und E/A-Geräten.	Läuft automatisch oder interaktiv. Verfügbar bei heruntergefahrenem Betriebssystem.	Lokal, kann aber mithilfe von ALOM angezeigt werden.
Solaris-Software-Befehle	Software	Anzeige verschiedener Systeminformationen.	Betriebssystem erforderlich.	Lokal, kann aber mithilfe von ALOM geöffnet werden.
SunVTS™-Software	Software	Systemprüfung und -belastung, Tests werden parallel ausgeführt.	Betriebssystem erforderlich. Optionales Paket.	Anzeige und Steuerung via Netzwerk.

Eingabeaufforderungen des Systems

Der Netra 240 Server verwendet die folgenden standardmäßigen Server-Eingabeaufforderungen:

- `ok` – OpenBoot PROM-Eingabeaufforderung
- `sc>` – Advanced Lights Out Manager- (ALOM-) Eingabeaufforderung
- `#` – Solaris Software-Superuser- (Bourne- und Korn-Shell-) Eingabeaufforderung

ABBILDUNG 1-1 zeigt die Beziehung zwischen den drei Eingabeaufforderungen und wie von einer Eingabeaufforderung zur anderen gewechselt wird.

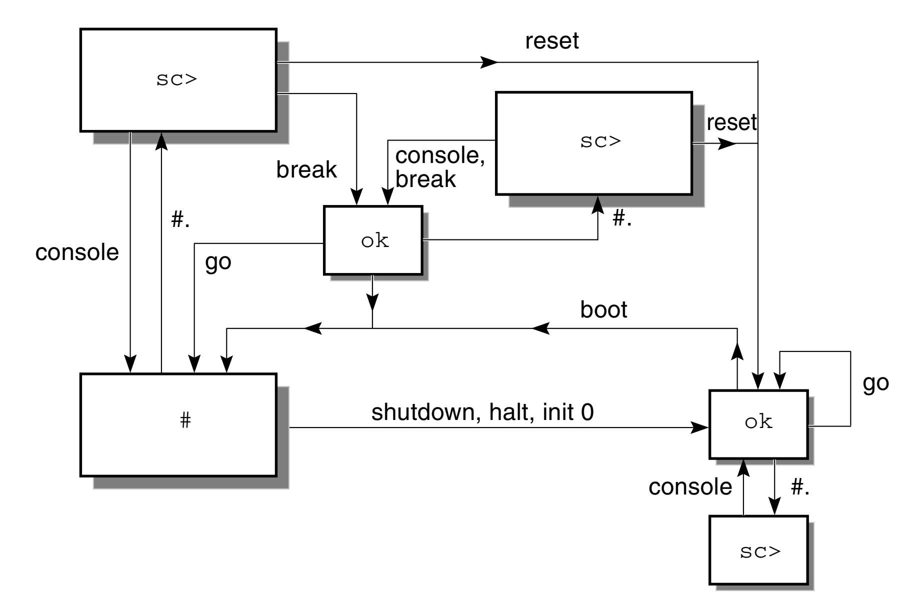


ABBILDUNG 1-1 Eingabeaufforderungs-Flussdiagramm

Die folgenden Befehle finden Sie im Flussdiagramm in ABBILDUNG 1-1:

- ALOM-Befehle: `console`, `reset`, `break`
- Escape-Folge: `#.`
- Solaris-Software-Befehle: `shutdown`, `halt`, `init 0`
- OpenBoot-Befehle: `go`, `boot`

Advanced Lights Out Manager

Sun™ Advanced Lights Out Manager (ALOM) für den Netra 240 Server bietet eine Reihe von LED-Statusanzeigen. In diesem Abschnitt wird ausführlich erläutert, was die Anzeigen jeweils bedeuten und wie sie ein- und ausgeschaltet werden. Weitere Informationen über ALOM finden Sie in Kapitel 3.

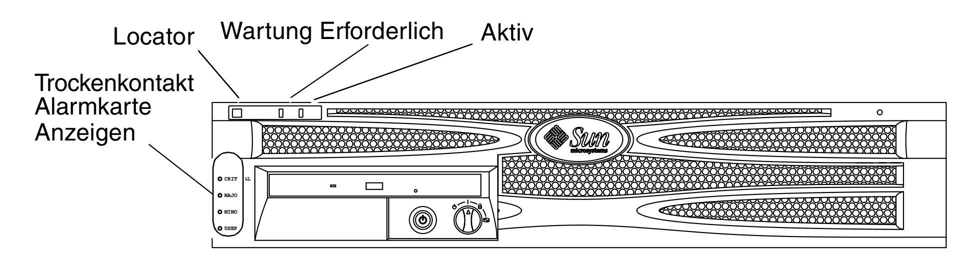


ABBILDUNG 1-2 Position der Anzeigen auf der Vorderseite

Serverstatusanzeigen

Der Server verfügt über drei LED-Statusanzeigen. Sie befinden sich auf dem Frontrahmen (ABBILDUNG 1-2) und auf der Rückseite. Eine Zusammenfassung der Anzeigen finden Sie in TABELLE 1-2.

TABELLE 1-2 Serverstatusanzeigen (auf Vorder- und Rückseite)

Anzeige	LED-Farbe	LED-Status	Bedeutung
Aktiv	Grün	Ein	Der Server wird gestartet und führt das Solaris-Betriebssystem aus.
		Aus	Entweder ist die Stromzufuhr nicht vorhanden oder das Solaris-Betriebssystem wird nicht ausgeführt.
Wartung erforderlich	Gelb	Ein	Beim Server liegt ein Problem vor, das vom Service-Personal behoben werden muss.
		Aus	Der Server hat keine Probleme erkannt.
Locator	Weiß	Ein	Eine kontinuierliche Anzeige leuchtet und kennzeichnet den Server in einem Rack, wenn der Befehl `setlocator` verwendet wird.

Sie können die Locator-LED über die Systemkonsole oder über die ALOM-Befehlszeilenschnittstelle (CLI) ein- und ausschalten.

So zeigen Sie den Locator-LED-Status an

- **Gehen Sie wie nachstehend beschrieben vor:**
  - **Wenn Sie als Superuser angemeldet sind, geben Sie Folgendes ein:**

```
# /usr/sbin/locator
```

  - **An der ALOM-Befehlszeilenschnittstelle hingegen müssen Sie die folgende Eingabe vornehmen:**

```
sc> showlocator
```

So schalten Sie die Locator-LED ein

- **Gehen Sie wie nachstehend beschrieben vor:**
  - **Wenn Sie als Superuser angemeldet sind, geben Sie Folgendes ein:**

```
# /usr/sbin/locator -n
```

  - **An der ALOM-Befehlszeilenschnittstelle hingegen müssen Sie die folgende Eingabe vornehmen:**

```
sc> setlocator on
```

So schalten Sie die Locator-LED aus

- **Gehen Sie wie nachstehend beschrieben vor:**
  - **Wenn Sie als Superuser angemeldet sind, geben Sie Folgendes ein:**

```
# /usr/sbin/locator -f
```

  - **An der ALOM-Befehlszeilenschnittstelle hingegen müssen Sie die folgende Eingabe vornehmen:**

```
sc> setlocator off
```

Alarmstatusanzeigen

Die Trockenkontakt-Alarmkarte hat vier LED-Statusanzeigen, die durch ALOM unterstützt werden. Sie sind senkrecht auf dem Frontrahmen angeordnet (ABBILDUNG 1-2). Informationen zu den Alarmanzeigen und den Trockenkontakt-Alarmzuständen finden Sie in TABELLE 1-3. Weitere Informationen über Alarmanzeigen finden Sie im *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch* (Teilenummer 817-5003-11). Weitere Informationen zur Steuerung der Alarmanzeigen mithilfe einer API finden Sie in Anhang A.

TABELLE 1-3 Alarmanzeigen- und Trockenkontaktalarmstatus

Anzeigen und Relay Bezeich-nungen	Anzeige-farbe	Anwendungs- oder Server-status	Bedingung oder Aktion	System-anzeige-status	Alarman-zeige-status	Relay NC[iv] Status	Relay NO[v] Status	Anmerkun-gen
Kritisch (Alarm0)	Rot	Serverstatus (einge-schaltet/ausgeschaltet und Solaris-Betriebs-system einsatzbereit/nicht einsatz-bereit)	Keine Strom-zufuhr.	Aus	Aus	Geschlos-sen	Geöffnet	Standard-status.
			System ausge-schaltet.	Aus	Aus[iii]	Geschlos-sen	Geöffnet	Eingangs-strom ange-schlossen.
			System einge-schaltet; Solaris-Betriebs-system nicht vollständig geladen.	Aus	Aus[iii]	Geschlos-sen	Geöffnet	Vorüberge-hender Status.
			Solaris-Betriebs-system erfolg-reich geladen.	Ein	Aus	Geöffnet	Geschlos-sen	Normaler Betriebs-status.

TABELLE 1-3 Alarmanzeigen- und Trockenkontaktalarmstatus *(Fortsetzung)*

Anzeigen und Relay Bezeichnungen	Anzeigefarbe	Anwendungs- oder Serverstatus	Bedingung oder Aktion	Systemanzeigestatus	Alarmanzeigestatus	Relay NC[iv] Status	Relay NO[v] Status	Anmerkungen
Kritisch (Alarm0) *(Forts.)*	Rot *(Forts.)*	Serverstatus (eingeschaltet/ausgeschaltet und Solaris-Betriebssystem einsatzbereit/ nicht einsatzbereit) *(Forts.)*	Watchdog-Timeout.	Aus	Ein	Geschlossen	Geöffnet	Vorübergehender Zustand; Neustart des Solaris-Betriebssystems.
			Herunterfahren des Solaris-Betriebssystems durch Benutzer initiiert[i].	Aus	Aus[iii]	Geschlossen	Geöffnet	Vorübergehender Status.
			Kein Eingangsstrom.	Aus	Aus	Geschlossen	Geöffnet	Standardstatus.
			Abschalten der Systemstromzufuhr durch Benutzer initiiert.	Aus	Aus[iii]	Geschlossen	Geöffnet	Vorübergehender Status.
		Anwendungsstatus	Benutzer aktiviert kritischen Alarm[ii].	—	Ein	Geschlossen	Geöffnet	Kritischer Fehler gefunden.
			Benutzer deaktiviert kritischen Alarm[ii].	—	Aus	Geöffnet	Geschlossen	Kritischer Fehler behoben.

TABELLE 1-3 Alarmanzeigen- und Trockenkontaktalarmstatus *(Fortsetzung)*

Anzeigen und Relay Bezeichnungen	Anzeigefarbe	Anwendungs- oder Serverstatus	Bedingung oder Aktion	Systemanzeigestatus	Alarmanzeigestatus	Relay NC[iv] Status	Relay NO[v] Status	Anmerkungen
Schwerwiegend (Alarm1)	Rot	Anwendungsstatus	Benutzer aktiviert schwerwiegenden Alarm[ii].	—	Ein	Geöffnet	Geschlossen	Schwerwiegender Fehler gefunden.
			Benutzer deaktiviert schwerwiegenden Alarm[ii].	—	Aus	Geschlossen	Geöffnet	Schwerwiegender Fehler behoben.
Geringfügiger (Alarm2)	Gelb	Anwendungsstatus	Benutzer aktiviert geringfügigen Alarm[ii].	—	Ein	Geöffnet	Geschlossen	Geringfügiger Fehler gefunden.
			Benutzer deaktiviert geringfügigen Alarm[ii].	—	Aus	Geschlossen	Geöffnet	Geringfügiger Fehler behoben.
Benutzer (Alarm3)	Gelb	Anwendungsstatus	Benutzer aktiviert Benutzeralarm[ii].	—	Ein	Geöffnet	Geschlossen	Benutzerfehler gefunden.
			Benutzer deaktiviert Benutzeralarm[ii].	—	Aus	Geschlossen	Geöffnet	Benutzerfehler behoben.

i Der Benutzer kann das System mithilfe von Befehlen wie `init0` und `init6` herunterfahren. Das System wird dabei nicht ausgeschaltet.

ii Basierend auf einer Festlegung der Fehlerbedingungen kann der Benutzer den Alarm mithilfe der Solaris-Plattformalarm-API oder ALOM CLI aktivieren. Weitere Informationen zur Alarm-API finden Sie in Anhang A, weitere Informationen zur ALOM CLI im *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch* (Teilenummer 817-5003-11).

iii Die Implementierung dieses Alarmanzeigestatus wird geändert.

iv Der NC-Status ist der normerweise geschlossene Status. Dieser Status stellt den Standardmodus der Relay-Kontakte im normalerweise geschlossenen Status dar.

v Der NO-Status ist der normalerweise geöffnete Status. Dieser Status stellt den Standardmodus der Relay-Kontakte im normalerweise geöffneten Status dar.

In allen Fällen, in denen ein Benutzer einen Alarm aktiviert, wird eine Meldung auf der Konsole angezeigt. Wenn beispielsweise der kritische Alarm aktiviert ist, wird die folgende Meldung auf der Konsole angezeigt:

```
SC Alert: CRITICAL ALARM is set
```

Beachten Sie, dass in manchen Fällen, in denen der kritische Alarm aktiviert ist, die zugehörige Alarmanzeige nicht leuchtet. Diese Implementierung wird in künftigen Versionen geändert (siehe Fußnote [iii] in TABELLE 1-3).

Power-On Self-Test-Diagnose (Einschaltselbsttest)

Power-on self-test (POST) ist ein Firmware-Programm, das festzustellen hilft, ob ein Teil des Systems ausgefallen ist. POST überprüft die Kernfunktionalität des Systems, einschließlich CPU-Modul(e), Hauptplatine, Speicher und einige On-Board-E/A-Geräte. Die Software generiert dann Meldungen, die bei der Bestimmung der Art eines Hardware-Fehlers nützlich sind. Sie können POST sogar ausführen, wenn das System nicht starten kann.

POST erkennt die meisten Systemfehler und befindet sich im OpenBoot PROM der Hauptplatine. Durch Festlegen von zwei Umgebungsvariablen können Sie die OpenBoot-Software so programmieren, dass POST beim Einschalten ausgeführt wird: die Flags `diag-switch?` und `diag-level`. Diese beiden Variablen sind auf der Systemkonfigurationskarte gespeichert.

POST wird automatisch ausgeführt, wenn das System eingeschaltet wird oder nach einem automatischen Zurücksetzen des Systems, wenn alle der folgenden Bedingungen vorliegen:

- `diag-switch?` wird auf `true` festgelegt (Standard ist `false`).
- `diag-level` wird auf `min`, `max` oder `menus` festgelegt (Standard ist `min`).
- `post-trigger` stimmt mit der Klasse von reset überein (Standard ist `power-on-reset`).

Wenn `diag-level` auf `min` oder `max` festgelegt wird, führt POST einen verkürzten bzw. erweiterten Test durch.

Wenn `diag-level` auf `menus` festgelegt ist, wird ein Menü aller Tests angezeigt, die beim Start ausgeführt werden.

POST-Diagnose- und Fehlermeldungsberichte werden auf einer Konsole angezeigt.

Steuern der POST-Diagnose

Die POST-Diagnose (und andere Aspekte des Boot-Prozesses) können Sie mit den OpenBoot-Konfigurationsvariablen steuern. Änderungen an der OpenBoot-Konfiguration werden erst nach einem Neustart des Systems wirksam. TABELLE 1-4 listet die wichtigsten und nützlichsten dieser Variablen auf. Anweisungen zur Einstellung der OpenBoot-Konfigurationsvariablen finden Sie in „So werden OpenBoot-Konfigurationsvariablen angezeigt und gesetzt" auf Seite 29.

TABELLE 1-4 OpenBoot-Konfigurationsvariablen

OpenBoot-Konfigurationsvariable	Beschreibung und Schlüsselwörter
`auto-boot`	Legt fest, ob das Betriebssystem automatisch gestartet wird. Die Standardeinstellung ist `true`. • `true` – Das Betriebssystem wird nach dem Ende der Firmware-Tests automatisch gestartet. • `false` – Die Eingabeaufforderung `ok` wird so lange angezeigt, bis Sie `boot` eingeben.
`diag-level`	Legt die Stufe bzw. den Typ der ausgeführten Tests fest. Die Standardeinstellung ist `min`. • `off` – Keine Tests. • `min` – Nur die grundlegenden Tests werden ausgeführt. • `max` – Je nach Gerät können umfassendere Tests ausgeführt werden. • `menus` – Menügesteuerte Tests können auf POST-Ebene einzeln ausgeführt werden.
`diag-script`	Legt fest, welche Geräte von den OpenBoot-Diagnoseroutinen untersucht werden. Die Standardeinstellung ist `none`. • `none` – Keine Geräte werden getestet. • `normal` – Geräte auf der Platine (basierend auf der Hauptplatine), für die es Selbsttests gibt, werden getestet. • `all` – Alle Geräte, für die es Selbsttests gibt, werden getestet.
`diag-switch?`	Schaltet den Diagnosemodus ein oder aus. Die Standardeinstellung ist `false`. • `true` – Diagnosemodus: POST-Diagnose und OpenBoot-Diagnoseprüfungen werden ausgeführt. • `false` – Standardmodus: Es werden keine POST- oder OpenBoot-Diagnoseprüfungen ausgeführt.

TABELLE 1-4 OpenBoot-Konfigurationsvariablen *(Fortsetzung)*

OpenBoot-Konfigurationsvariable	Beschreibung und Schlüsselwörter
`post-trigger` `obdiag-trigger`	Diese beiden Variablen legen die Klasse des Reset-Ereignisses fest, das Einschaltselbsttests (oder die Tests der OpenBoot-Diagnoseprüfungen) ausführt. Als Variablen können einzelne Schlüsselwörter oder Kombinationen der ersten drei Schlüsselwörter jeweils getrennt durch ein Leerzeichen eingegeben werden. Weitere Informationen finden Sie unter „So werden OpenBoot-Konfigurationsvariablen angezeigt und gesetzt“ auf Seite 29. • `error-reset` – Ein von bestimmten irreparablen Hardwarefehlern hervorgerufenes Reset. In der Regel kommt es zu einem Fehler-Reset, wenn die Systemdaten durch ein Hardwareproblem beschädigt wurden. Beispiele für einen solchen Fehler sind CPU- und System-Watchdog-Resets, schwerwiegende Fehler und bestimmte CPU-Reset-Ereignisse (Standard). • `power-on-reset` – Ein Reset, das durch Drücken des Ein-/Standby-Schalters hervorgerufen wird (Standard). • `user-reset` – Ein vom Benutzer oder Betriebssystem veranlasstes Reset. • `all-resets` – Jede Art von System-Reset. • `none` – Es werden keine Einschaltselbsttests (oder OpenBoot-Diagnoseprüfungen) ausgeführt.
`input-device`	Legt fest, woher die Konsoleneingabe stammt. Die Standardeinstellung ist `ttya`. • `ttya` – Vom integrierten SERIAL MGT-Anschluss. • `ttya` – Vom integrierten allgemeinen Anschluss (10101). • `keyboard` – Von der angeschlossenen Tastatur, die Teil eines Grafikterminals ist.
`output-device`	Legt fest, wohin Diagnose- und andere Konsolenmeldungen ausgegeben werden. Die Standardeinstellung ist `ttya`. • `ttya` – Zum integrierten SERIAL MGT-Anschluss. • `ttya` – Zum integrierten allgemeinen Anschluss (10101). • `screen` – Zum angeschlossenen Bildschirm, der Teil des Grafikterminals ist.[i]

i POST-Meldungen können nicht auf einem Grafikterminal angezeigt werden. Sie werden an `ttya` ausgegeben, selbst wenn `output-device` auf screen gesetzt ist.

Hinweis – Diese Variablen beeinflussen sowohl die OpenBoot-Diagnoseprüfungen als auch die POST-Diagnose.

Nach Beendigung der POST-Diagnose meldet POST den Status jedes einzelnen Tests an die OpenBoot-Firmware. Die Kontrolle geht danach zurück an den OpenBoot-Firmware-Code.

Wenn die POST-Diagnose einen Fehler nicht finden kann und der Server noch immer nicht startet, führen Sie die OpenBoot-Diagnoseprüfungen durch.

▼ So starten Sie die POST-Diagnose

1. **Wechseln Sie zur Eingabeaufforderung** `ok`**.**
2. **Geben Sie die folgende Zeile ein:**

```
ok setenv diag-switch? true
```

3. **Geben Sie die folgende Zeile ein:**

```
ok setenv diag-level Wert
```

   Wobei *Wert* entweder `min`, `max` oder `menus` ist, abhängig von der Menge der Diagnosedaten, die Sie anzeigen möchten.

4. **Geben Sie die folgende Zeile ein:**

```
ok reset-all
```

   Das System führt die POST-Diagnose aus, wenn `post-trigger` auf `user-reset` festgelegt ist. Status- und Fehlermeldungen werden im Konsolenfenster angezeigt. Wenn POST einen Fehler findet, wird eine Fehlermeldung mit einer Beschreibung des Fehlers angezeigt.

5. **Wenn Sie die Ausführung von POST abgeschlossen haben, setzen Sie den Wert von** `diag-switch?` **auf** `false` **zurück, indem Sie Folgendes eingeben:**

```
ok setenv diag-switch? false
```

   Wenn Sie `diag-switch?` auf `false` zurücksetzen, wird die Startzeit minimiert.

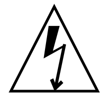
OpenBoot-Befehle

OpenBoot-Befehle sind Befehle, die Sie in der Eingabeaufforderung `ok` eingeben. Folgende OpenBoot-Befehle können nützliche Diagnoseinformationen liefern:

- `probe-scsi` und `probe-scsi-all`
- `probe-ide`
- `show-devs`

Die Befehle `probe-scsi` und `probe-scsi-all`

Die Befehle `probe-scsi` und `probe-scsi-all` untersuchen Probleme mit SCSI-Geräten.



Achtung – Wenn Sie die Eingabeaufforderung `ok` mit dem Befehl `halt` oder der Tastenfolge Stop-A angezeigt haben und danach `probe-scsi` oder `probe-scsi-all` eingeben, kann das System hängen bleiben.

Der Befehl `probe-scsi` kommuniziert mit allen SCSI-Geräten, die an On-Board-SCSI-Controller angeschlossen sind. Der Befehl `probe-scsi-all` greift auch auf alle Geräte zu, die an Hostadapter in PCI-Steckplätzen angeschlossen sind.

`probe-scsi` und `probe-scsi-all` zeigen für alle angeschlossenen und aktiven SCSI-Geräte die Loop-ID, den Hostadapter, die Nummer der logischen Einheit, den eindeutigen World Wide Name (WWN) sowie eine Gerätebeschreibung mit Typ und Hersteller an.

Folgendes Beispiel ist eine Ausgabe des Befehls `probe-scsi`.

CODE-BEISPIEL 1-1 Ausgabe des Befehls `probe-scsi`

```
{1} ok probe-scsi
Target 0
  Unit 0   Disk     SEAGATE ST373307LSUN72G 0207
Target 1
  Unit 0   Disk     SEAGATE ST336607LSUN36G 0207
{1} ok
```

Das folgende Beispiel ist eine Ausgabe des Befehls `probe-scsi-all`.

CODE-BEISPIEL 1-2 Ausgabe des Befehls `probe-scsi-all`

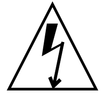
```
{1} ok probe-scsi-all
/pci@1c,600000/scsi@2,1

/pci@1c,600000/scsi@2
Target 0
  Unit 0   Disk     SEAGATE ST373307LSUN72G 0207
Target 1
  Unit 0   Disk     SEAGATE ST336607LSUN36G 0207

{1} ok
```

Der Befehl `probe-ide`

Der Befehl `probe-ide` kommuniziert mit allen IDE (Integrated Drive Electronics)-Geräten, die am IDE-Bus angeschlossen sind. Dies ist der interne Systembus für Speichergeräte wie das DVD-Laufwerk.



Achtung – Wenn Sie die Eingabeaufforderung `ok` mit dem Befehl `halt` oder der Tastenfolge Stop-A angezeigt haben und danach `probe-ide` eingeben, kann das System hängen bleiben.

Folgendes Beispiel ist eine Ausgabe des Befehls `probe-ide`.

CODE-BEISPIEL 1-3 Ausgabe des Befehls `probe-ide`

```
{1} ok probe-ide
Device 0  ( Primary Master )
         Not Present

  Device 1  ( Primary Slave )
         Not Present

  Device 2  ( Secondary Master )
         Not Present

  Device 3  ( Secondary Slave )
         Not Present

{1} ok
```

Der Befehl `show-devs`

Mit dem Befehl `show-devs` werden die Hardware-Gerätepfade aller Geräte des Firmware-Gerätebaums aufgelistet. CODE-BEISPIEL 1-4 zeigt eine Beispielausgabe.

CODE-BEISPIEL 1-4 Ausgabe des Befehls `show-devs`

```
/pci@1d,700000
/pci@1c,600000
/pci@1e,600000
/pci@1f,700000
/memory-controller@1,0
/SUNW,UltraSPARC-IIIi@1,0
/memory-controller@0,0
/SUNW,UltraSPARC-IIIi@0,0
/virtual-memory
/memory@m0,0
/aliases
/options
/openprom
/chosen
/packages
/pci@1d,700000/network@2,1
/pci@1d,700000/network@2
/pci@1c,600000/scsi@2,1
/pci@1c,600000/scsi@2
/pci@1c,600000/scsi@2,1/tape
/pci@1c,600000/scsi@2,1/disk
/pci@1c,600000/scsi@2/tape
/pci@1c,600000/scsi@2/disk
/pci@1e,600000/ide@d
/pci@1e,600000/usb@a
/pci@1e,600000/pmu@6
/pci@1e,600000/isa@7
/pci@1e,600000/ide@d/cdrom
/pci@1e,600000/ide@d/disk.........
```

▼ So führen Sie OpenBoot-Befehle aus

1. **Halten Sie das System an, um zur Eingabeaufforderung `ok` zu gelangen.**

   Informieren Sie die Benutzer, bevor Sie das System herunterfahren.

2. **Geben Sie den entsprechenden Befehl an der Eingabeaufforderung der Konsole ein.**

OpenBoot-Diagnoseroutinen

Wie die POST-Diagnose basiert der Code der OpenBoot-Diagnoseroutinen auf Firmware und ist in Boot PROM resident.

▼ So starten Sie OpenBoot-Diagnoseprüfungen

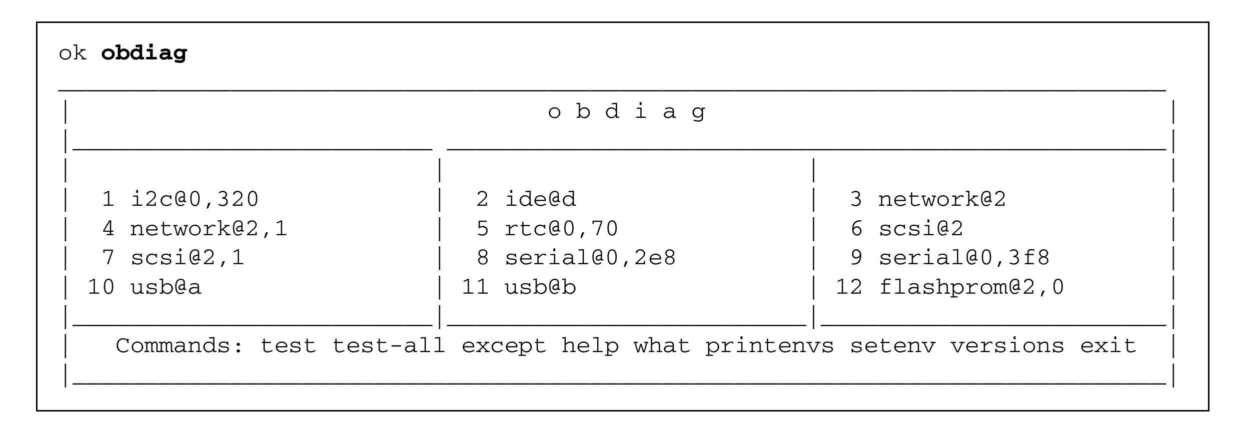
1. **Typ:**

```
ok setenv diag-switch? true
ok setenv auto-boot? false
ok reset-all
```

2. **Typ:**

```
ok obdiag
```

Dieser Befehl zeigt das OpenBoot-Diagnoseroutinenmenü an.



```
ok obdiag
_____________________________________________________________________________
|                                 o b d i a g                                 |
|_________________________ _________________________________________________|
|                         |                         |                         |
|  1 i2c@0,320            |  2 ide@d                |  3 network@2            |
|  4 network@2,1          |  5 rtc@0,70             |  6 scsi@2               |
|  7 scsi@2,1             |  8 serial@0,2e8         |  9 serial@0,3f8         |
| 10 usb@a                | 11 usb@b                | 12 flashprom@2,0        |
|_________________________|_________________________|_________________________|
|   Commands: test test-all except help what printenvs setenv versions exit   |
|_____________________________________________________________________________|
```

Hinweis – Wenn Sie im Server eine PCI-Karte installiert haben, werden zusätzliche Prüfungen im Menü `obdiag` angezeigt.

---

3. Geben Sie die folgende Zeile ein:

```
obdiag> test n
```

Dabei steht *n* für die Nummer des jeweiligen Tests, den Sie ausführen möchten.

Eine Zusammenfassung der Prüfungen ist verfügbar. Geben Sie an der Eingabeaufforderung `obdiag>` Folgendes ein:

```
obdiag> help
```

Steuerung der OpenBoot-Diagnoseprüfungen

Die meisten OpenBoot-Konfigurationsvariablen, mit denen Sie die POST-Tests steuern (siehe TABELLE 1-4), beeinflussen auch die Ausführung der OpenBoot Diagnostics-Tests.

- Verwenden Sie die Variable `diag-level`, um die Teststufe der OpenBoot-Diagnoseprüfungen zu steuern.
- Verwenden Sie `test-args`, um die Art der Testausführung anzupassen.

Standardmäßig enthält die Variable `test-args` eine Leerzeichenkette. Sie können `test-args` ändern, indem Sie eins oder mehrere der reservierten Schlüsselwörter aus TABELLE 1-5 verwenden.

TABELLE 1-5 Schlüsselwörter für die OpenBoot-Konfigurationsvariable `test-args`

Schlüsselwort	Beschreibung
`bist`	Startet den integrierten Selbsttest (Built-in Self-Test, BIST) für externe und Peripheriegeräte.
`debug`	Zeigt alle Fehlermeldungen an.
`iopath`	Prüft die Bus- und Interconnect-Integrität.
`loopback`	Prüft den externen Loopback-Pfad des Geräts.
`media`	Prüft den Zugriff auf die Medien externer Geräte und Peripheriegeräte.
`restore`	Versucht, den Originalzustand des Geräts wiederherzustellen, falls die vorherige Testausführung fehlgeschlagen ist.
`silent`	Zeigt nur Fehler an, nicht aber den Status der einzelnen Tests.
`subtests`	Zeigt den Haupttest und alle aufgerufenen Untertests an.

TABELLE 1-5 Schlüsselwörter für die OpenBoot-Konfigurationsvariable `test-args` *(Fortsetzung)*

Schlüsselwort	Beschreibung
`verbose`	Zeigt ausführliche Statusmeldungen zu allen Tests an.
`callers=`*n*	Zeigt bei einem Fehler eine Liste der letzten *n* Aufrufer an: `callers=0` – zeigt eine Liste aller Aufrufer vor dem Fehler an.
`errors=`*n*	Beendet den Test nach *n* Fehlern: `errors=0` – zeigt alle Fehlermeldungen an, ohne den Test zu beenden.

Wenn Sie die OpenBoot-Diagnoseprüfungen anpassen möchten, können Sie für `test-args` eine komma-getrennte Schlüsselwortliste einsetzen. Zum Beispiel:

```
ok setenv test-args debug,loopback,media
```

Die Befehle `test` und `test-all`

Die OpenBoot-Diagnoseprüfungen können auch direkt von der Eingabeaufforderung `ok` ausgeführt werden. Dazu geben Sie den Befehl `test` gefolgt vom vollständigen Hardwarepfad des bzw. der zu prüfenden Geräte ein. Beispiel:

```
ok test /pci@x,y/SUNW,qlc@2
```

Einzelne Tests können Sie mit dem Befehl `test-args` wie folgt einstellen:

```
ok test /usb@1,3:test-args={verbose,debug}
```

Diese Syntax betrifft nur den aktuellen Test, ohne dass der Wert der OpenBoot-Konfigurationsvariable `test-args` geändert wird.

Alle Geräte des Gerätebaums können mit dem Befehl `test-all` geprüft werden:

```
ok test-all
```

Wenn Sie für `test-all` ein Pfadargument angeben, werden nur das betreffende Gerät und seine untergeordneten Komponenten geprüft. Mit dem folgenden Befehl werden beispielsweise der USB-Bus und alle Geräte geprüft, die mit dem USB-Bus verbunden sind und über einen Selbsttest verfügen:

```
ok test-all /pci@9,700000/usb@1,3
```

Fehlermeldungen der OpenBoot-Diagnoseroutinen

Fehler, die sich während der Ausführung der OpenBoot-Diagnoseroutinen ergeben, werden in tabellarischer Form mit einem kurzen Überblick des Problems, dem betroffenen Hardwaregerät, dem nicht bestandenen Untertest sowie weiteren Diagnoseinformationen zurückgegeben. CODE-BEISPIEL 1-5 ist ein Beispiel für eine Fehlermeldung der OpenBoot-Diagnoseprüfungen.

CODE-BEISPIEL 1-5 Fehlermeldung der OpenBoot-Diagnoseroutinen

```
Testing /pci@1e,600000/isa@7/flashprom@2,0

   ERROR: FLASHPROM CRC-32 is incorrect
   SUMMARY: Obs=0x729f6392 Exp=0x3d6cdf53 XOR=0x4ff3bcc1 Addr=0xfeebbffc
   DEVICE  : /pci@1e,600000/isa@7/flashprom@2,0
   SUBTEST : selftest:crc-subtest
   MACHINE : Netra 240
   SERIAL# : 52965531
   DATE    : 03/05/2003 01:33:59  GMT
   CONTR0LS: diag-level=max test-args=

Error: /pci@1e,600000/isa@7/flashprom@2,0 selftest failed, return code = 1
Selftest at /pci@1e,600000/isa@7/flashprom@2,0 (errors=1) ............
fehlgeschlagen
Pass:1 (of 1) Errors:1 (of 1) Tests Failed:1 Elapsed Time: 0:0:0:27
```

Betriebssystem-Diagnosewerkzeuge

Wenn ein System die OpenBoot-Diagnoseprüfungen besteht, versucht es das Solaris-Betriebssystem zu starten. Sobald der Server im Mehrbenutzermodus läuft, haben Sie Zugriff auf Software-basierte Diagnosewerkzeuge und die SunVTS-Software. Mit diesen Werkzeugen können Sie den Server überwachen, testen und Fehler isolieren.

Hinweis – Wenn Sie die OpenBoot-Konfigurationsvariable `auto-boot?` auf `false` setzen, wird das Betriebssystem nach Beendigung der firmwarebasierten Tests *nicht* gestartet.

Neben den oben genannten Werkzeugen stehen Ihnen Fehler- und System-meldungsprotokolldateien sowie Solaris Softwareinformationsbefehle zur Verfügung.

Protokolldateien für Fehler- und Systemmeldungen

Fehler- und andere Systemmeldungen werden in der Datei `/var/adm/messages` gespeichert. Die in diese Datei geschriebenen Meldungen stammen von mehreren Systemkomponenten wie dem Betriebssystem, der Umgebungsüberwachung und verschiedenen Softwareanwendungen.

Befehle zur Abfrage von Solaris Softwaresysteminformationen

Die folgenden Solaris Software-Systeminformationen zeigen Daten an, die Sie zur Bewertung des Zustands eines Netra 240 Servers verwenden können.

- `prtconf`
- `prtdiag`
- `prtfru`
- `psrinfo`
- `showrev`

Die von diesen Befehlen zurückgegebenen Informationen werden in diesem Abschnitt beschrieben. Weitere Informationen zur Verwendung dieser Befehle finden Sie auf der entsprechenden Man Page.

Der Befehl `prtconf`

Mit dem Befehl `prtconf` wird der Solaris Software-Gerätebaum angezeigt. Dieser Baum enthält neben den von der OpenBoot-Firmware geprüften Geräten weitere Geräte wie einzelne Festplatten, die nur die Software des Betriebssystems kennt. Außerdem wird mit `prtconf` der Gesamtsystemspeicher angezeigt. CODE-BEISPIEL 1-6 ist ein Ausschnitt aus der Ausgabe von `prtconf`.

CODE-BEISPIEL 1-6 Ausgabe des Befehls `prtconf`

```
# prtconf

System Configuration:  Sun Microsystems sun4u
Memory size: 5120 Megabytes
System Peripherals (Software Nodes):

SUNW,Netra-240
    packages (driver not attached)
        SUNW,builtin-drivers (driver not attached)
        deblocker (driver not attached)
        disk-label (driver not attached)
        terminal-emulator (driver not attached)
        dropins (driver not attached)
        kbd-translator (driver not attached)
        obp-tftp (driver not attached)
        SUNW,i2c-ram-device (driver not attached)
        SUNW,fru-device (driver not attached)
        ufs-file-system (driver not attached)
    chosen (driver not attached)
    openprom (driver not attached)
        client-services (driver not attached)
    options, instance #0
    aliases (driver not attached)
    memory (driver not attached)
    virtual-memory (driver not attached)
    SUNW,UltraSPARC-IIIi (driver not attached)
    memory-controller, instance #0
    SUNW,UltraSPARC-IIIi (driver not attached)
    memory-controller, instance #1
    pci, instance #0........
```

Mit der Option `-p` gibt der Befehl `prtconf` ähnliche Informationen zurück wie der OpenBoot-Befehl `show-devs`. Diese Ausgabe führt nur die von der System-Firmware kompilierten Geräte auf.

Der Befehl prtdiag

Der Befehl prtdiag zeigt eine Tabelle mit Diagnoseinformationen über den Status der Systemkomponenten an. Das Anzeigeformat des Befehls prtdiag hängt von dem auf dem System laufenden Solaris-Betriebssystem ab. Das folgende Codebeispiel ist ein Ausschnitt der Ausgabe, die durch prtdiag auf einem betriebsbereiten Netra 240 erstellt wird, auf dem die Solaris Software ausgeführt wird.

CODE-BEISPIEL 1-7 Ausgabe des Befehls prtdiag

```
# prtdiag
System Configuration: Sun Microsystems  sun4u Netra 240
System clock frequency: 160 MHZ
Memory size: 2GB
==================================== CPUs ====================================
                     E$          CPU     CPU      Temperature        Fan
      CPU  Freq      Size        Impl.   Mask     Die    Ambient   Speed   Unit
      ---  --------  ----------  ------  ----  --------  --------  -----   ----
     MB/P0  1280 MHz  1MB         US-IIIi   2.3     -     -
     MB/P1  1280 MHz  1MB         US-IIIi   2.3     -     -
================================= IO Devices =================================
     Bus   Freq
Brd  Type  MHz   Slot        Name                          Model
---  ----  ----  ----------  ----------------------------  --------------------
 0   pci    66            2  network-pci14e4,1648.108e.16+
 0   pci    66            2  network-pci14e4,1648.108e.16+
 0   pci    66            2  scsi-pci1000,21.1000.1000.1 +
 0   pci    66            2  scsi-pci1000,21.1000.1000.1 +
 0   pci    66            2  network-pci14e4,1648.108e.16+
 0   pci    66            2  network-pci14e4,1648.108e.16+
 0   pci    33            7  isa/serial-su16550 (serial)
 0   pci    33            7  isa/serial-su16550 (serial)
 0   pci    33            7  isa/rmc-comm-rmc_comm (seria+
 0   pci    33           13  ide-pci10b9,5229.c4 (ide)
============================= Memory Configuration ============================
Segment Table:
-----------------------------------------------------------------------
Base Address       Size       Interleave Factor  Contains
-----------------------------------------------------------------------
0x0                1GB               1           GroupID 0
0x1000000000       1GB               1           GroupID 0
```

CODE-BEISPIEL 1-7 Ausgabe des Befehls `prtdiag` *(Fortsetzung)*

```
Memory Module Groups:
--------------------------------------------------
ControllerID   GroupID  Labels
--------------------------------------------------
0              0        MB/P0/B0/D0,MB/P0/B0/D1
Memory Module Groups:
--------------------------------------------------
ControllerID   GroupID  Labels
--------------------------------------------------
1              0        MB/P1/B0/D0,MB/P1/B0/D1
```

Neben den Informationen aus CODE-BEISPIEL 1-7 gibt `prtdiag` mit der ausführlichen Option (`-v`) zusätzlich den Status der Frontplatte, der Festplatten, der Lüfter, der Netzteile sowie Hardware-Versionen und Systemtemperaturen zurück (siehe CODE-BEISPIEL 1-8).

CODE-BEISPIEL 1-8 Ausgabe des Befehls `prtdiag` mit der Option -v

```
---------------------------------------------------------------
Location    Sensor      Temperature  Lo LoWarn HiWarn  Hi Status
---------------------------------------------------------------
MB          T_ENC           22C     -7C   -5C    55C   58C   okay
MB/P0       T_CORE          57C     -        -    110C  115C okay
MB/P1       T_CORE          54C     -        -    110C  115C okay
PS0         FF_OT           -       -        -    -        -  okay
PS1         FF_OT           -       -        -    -        -  okay
```

Bei Überhitzung gibt `prtdiag` in der Status-Spalte einen Fehler zurück (CODE-BEISPIEL 1-9).

CODE-BEISPIEL 1-9 Ausgabe des Befehls `prtdiag` mit Überhitzungsanzeige

```
---------------------------------------------------------------
Location    Sensor      Temperature  Lo LoWarn HiWarn  Hi Status
---------------------------------------------------------------
MB          T_ENC           22C     -7C   -5C    55C   58C   okay
MB/P0       T_CORE         118C     -        -    110C  115C failed
MB/P1       T_CORE         112C    -        -    110C  115C warning
PS0         FF_OT           -       -        -    -        -  okay
PS1         FF_OT           -       -        -    -        -  okay
```

Auch beim Ausfall einer bestimmten Komponente meldet `prtdiag` einen Fehler in der entsprechenden Statusspalte (CODE-BEISPIEL 1-10).

CODE-BEISPIEL 1-10 Ausgabe des Befehls `prtdiag` mit Fehleranzeige

```
Lüftergeschwindigkeiten:
-----------------------------------------
Location        Sensor      Status   Speed
-----------------------------------------
MB/P0/F0        RS          failed   0 rpm
MB/P0/F1        RS          okay     3994 rpm
F2              RS          okay     2896 rpm
PS0             FF_FAN      okay
F3              RS          okay     2576 rpm
PS1             FF_FAN      okay
---------------------------------
```

Der Befehl `prtfru`

Der Netra 240 Server zeichnet alle FRUs des Systems sowie spezielle Informationen über die einzelnen FRUs in einer hierarchischen Liste auf.

Der Befehl `prtfru` kann sowohl diese hierarchische Liste wie auch die auf den SEEPROM (Serial Electrically-Erasable Programmable Read-Only Memory)-Geräten vieler FRUs gespeicherten Daten anzeigen. CODE-BEISPIEL 1-11 ist ein Ausschnitt einer hierarchischen FRU-Liste, die mit der Option `-l` des Befehls `prtfru` generiert wurde.

CODE-BEISPIEL 1-11 Ausgabe des Befehls `prtfru -l`

```
# prtfru -l
/frutree
/frutree/chassis (fru)
/frutree/chassis/MB?Label=MB
/frutree/chassis/MB?Label=MB/system-board (container)
/frutree/chassis/MB?Label=MB/system-board/SC?Label=SC
/frutree/chassis/MB?Label=MB/system-board/SC?Label=SC/sc (fru)
/frutree/chassis/MB?Label=MB/system-board/BAT?Label=BAT
/frutree/chassis/MB?Label=MB/system-board/BAT?Label=BAT/battery (fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu (fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F0?Label=F0
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F0?Label=F0/fan-unit
(fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F1?Label=F1
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F1?Label=F1/fan-unit
(fru)........
```

CODE-BEISPIEL 1-12 ist ein Ausschnitt der von `prtfru -c` generierten SEEPROM-Daten. Diese Ausgabe zeigt nur die Container und ihre Daten an, druckt jedoch nicht die FRU-Baumhierarchie.

CODE-BEISPIEL 1-12 Ausgabe des Befehls `prtfru -c`

```
# prtfru -c
/frutree/chassis/MB?Label=MB/system-board (container)
   SEGMENT: SD
      /ManR
      /ManR/UNIX_Timestamp32: Mon Dec  2 19:47:38 PST 2002
      /ManR/Fru_Description: FRUID,INSTR,M'BD,2X1.28GHZ,CPU
      /ManR/Manufacture_Loc: Hsinchu,Taiwan
      /ManR/Sun_Part_No: 3753120
      /ManR/Sun_Serial_No: 000615
      /ManR/Vendor_Name: Mitac International
      /ManR/Initial_HW_Dash_Level: 02
      /ManR/Initial_HW_Rev_Level: 0E
      /ManR/Fru_Shortname: MOTHERBOARD
      /SpecPartNo: 885-0076-11
/frutree/chassis/MB?Label=MB/system-board/P0?Label=
P0/cpu/B0?Label=B0/bank/D0?La
bel=D0/mem-module (container)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=
P0/cpu/B0?Label=B0/bank/D1?La
bel=D1/mem-module (container)........
```

Die von `prtfru` angezeigten Daten variieren je nach FRU-Typ. In der Regel werden folgende Informationen angezeigt:

- FRU-Beschreibung
- Herstellername und -ort
- Teilenummer und Seriennummer
- Hardwareversion

Der Befehl `psrinfo`

Der Befehl `psrinfo` zeigt das Datum und die Uhrzeit an, an dem die einzelnen CPUs online geschaltet wurden. Mit der ausführlichen Option (`-v`) zeigt der Befehl zusätzliche Informationen zu den CPUs an, einschließlich ihrer Taktfrequenz. CODE-BEISPIEL 1-13 zeigt ein Beispiel für die Ausgabe des Befehls `psrinfo` mit der Option `-v`.

CODE-BEISPIEL 1-13 Ausgabe des Befehls `psrinfo -v`

```
# psrinfo -v
Status of processor 0 as of: 07/28/2003 14:43:29
  Processor has been on-line since 07/21/2003 18:43:37.
  The sparcv9 processor operates at 1280 MHz,
        and has a sparcv9 floating point processor.
Status of processor 1 as of: 07/28/2003 14:43:29
  Processor has been on-line since 07/21/2003 18:43:36.
  The sparcv9 processor operates at 1280 MHz,
        and has a sparcv9 floating point processor
```

Der Befehl showrev

Mit dem Befehl `showrev` werden die Versionen der aktuellen Hardware- und Softwarekomponenten angezeigt. CODE-BEISPIEL 1-14 ist eine Beispielausgabe des Befehls `showrev`.

CODE-BEISPIEL 1-14 Ausgabe des Befehls `showrev`

```
# showrev
Hostname: vsp78-36
Hostid: 8328c87b
Release: 5.8
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: vsplab.SFBay.Sun.COM
Kernel version: SunOS 5.8 Generic 108528-18 November 2002
```

Mit der Option `-p` zeigt der Befehl `showrev` die installierten Patches an. CODE-BEISPIEL 1-15 zeigt eine teilweise Beispielsausgabe des Befehls `showrev` mit der Option `-p`.

CODE-BEISPIEL 1-15 Ausgabe des Befehls `showrev -p`

```
Patch: 109729-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 109783-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 109807-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 109809-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 110905-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 110910-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 110914-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 108964-04 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsr
```

▼ So führen Sie Befehle zur Abfrage von Solaris Plattformsysteminformationen aus

- **Geben Sie an einer Befehlseingabeaufforderung den Befehl für die Art der Systeminformationen ein, die Sie anzeigen möchten.**

  Weitere Informationen zu diesem Thema finden Sie unter „Befehle zur Abfrage von Solaris Softwaresysteminformationen“ auf Seite 20. TABELLE 1-6 enthält einen Überblick über die Befehle.

TABELLE 1-6 Solaris Plattform-Informationsanzeigebefehle

Befehl	Anzeige	Erforderliche Eingabe	Anmerkung
`prtconf`	Informationen zur System-konfiguration	`/usr/sbin/prtconf`	—
`prtdiag`	Diagnose- und Konfigurations-informationen	`/usr/platform/sun4u/sbin/prtdiag`	Weitere Einzelheiten erhalten Sie mit der Option `-v`.
`prtfru`	FRU-Hierarchie und SEEPROM-Speicherinhalt	`/usr/sbin/prtfru`	Mit der Option `-l` zeigen Sie die Hierarchie an. Mit der Option `-c` zeigen Sie die SEEPROM-Daten an.
`psrinfo`	Datum und Uhrzeit der Inbetriebnahme der einzelnen CPUs; Prozessortaktfrequenz	`/usr/sbin/psrinfo`	Mit der Option `-v` rufen Sie die Taktfrequenz sowie weitere Daten ab.
`showrev`	Revisionsinformationen zu Hardware und Software	`/usr/bin/showrev`	Mit der Option `-p` zeigen Sie Software-Patches an.

Die neuesten Diagnoseprüfungsergebnisse

Eine Zusammenfassung der Ergebnisse der letzten POST- und OpenBoot-Diagnoseprüfungen bleibt auch nach dem Abschalten gespeichert.

▼ So zeigen Sie die neuesten Prüfungsergebnisse an

1. **Wechseln Sie zur Eingabeaufforderung `ok`.**
2. **Führen Sie eine der folgenden Maßnahmen durch:**
   - Mit dem folgenden Befehl zeigen Sie eine Zusammenfassung der letzten POST-Ergebnisse an:

```
ok show-post-results
```

   - Geben Sie die folgende Zeile ein, um eine Zusammenfassung der letzten OpenBoot-Diagnoseprüfungsergebnisse anzuzeigen:

```
ok show-obdiag-results
```

Mit diesem Befehl wird eine systemabhängige Liste der Hardwarekomponenten erstellt, der Sie entnehmen können, welche Bauteile die POST- bzw. OpenBoot-Diagnoseprüfungen bestanden haben und welche nicht.

OpenBoot-Konfigurationsvariablen

Schalter und Diagnosekonfigurationsvariablen, die im IDPROM gespeichert sind, bestimmen, wie und wann POST-Diagnoseroutinen und OpenBoot-Diagnosetests ausgeführt werden. Nachfolgend erfahren Sie, wie Sie auf die OpenBoot-Konfigurationsvariablen zugreifen, um sie zu ändern. Eine Liste wichtiger OpenBoot-Konfigurationsvariablen finden Sie in TABELLE 1-4.

Die Änderungen, die Sie an OpenBoot-Konfigurationsvariablen vornehmen, werden erst nach dem nächsten Neustart wirksam.

▼ So werden OpenBoot-Konfigurationsvariablen angezeigt und gesetzt

- **Halten Sie den Server an, um die Eingabeaufforderung `ok` anzuzeigen.**
  - Zeigen Sie die aktuellen Werte aller OpenBoot-Konfigurationsvariablen mit dem Befehl `printenv` an.

    Das nachstehende Beispiel enthält einen kurzen Auszug der Ausgabe dieses Befehls.

```
ok printenv
Variable Name         Value                          Default Value

diag-level            min                             min
diag-switch?          false                           false
```

  - Den Wert einer OpenBoot-Konfigurationsvariable setzen bzw. ändern Sie mit dem Befehl `setenv`:

```
ok setenv diag-level max
diag-level =          max
```

  - Um OpenBoot-Konfigurationsvariablen festzulegen, die mehrere Schlüsselwörter akzeptieren, müssen Sie die Schlüsselwörter durch ein Leerzeichen voneinander trennen.

Verwenden der Befehle `watch-net` und `watch-net-all` zum Prüfen der Netzwerkverbindungen

Die Diagnoseprüfung `watch-net` überwacht Ethernet-Pakete auf der primären Netzwerkschnittstelle. Die Diagnoseprüfung `watch-net-all` überwacht Ethernet-Pakete auf der primären Netzwerkschnittstelle und auf etwaigen zusätzlichen Netzwerkschnittstellen, die an die Systemplatine angeschlossen sind. Gute vom System empfangene Pakete werden durch einen Punkt (.) gekennzeichnet. Fehler, wie Rahmenfehler oder CRC-Fehler (cyclic redundancy check) werden durch ein X und eine zugehörige Fehlerbeschreibung gekennzeichnet.

- **Starten Sie den Diagnosetest** `watch-net`**, indem Sie den Befehl** `watch-net` **bei der Eingabeaufforderung** `ok` **eingeben (**CODE-BEISPIEL 1-16**).**

CODE-BEISPIEL 1-16 Diagnoseausgabemeldung `watch-net`

```
{0} ok watch-net
Internal loopback test -- succeeded.
Link is -- up
Looking for Ethernet Packets.
'.' is a Good Packet. 'X' is a Bad Packet.
Type any key to stop................................
```

- **Starten Sie den Diagnosetest** `watch-net-all`**, indem Sie den Befehl** `watch-net-all` **bei der Eingabeaufforderung** `ok` **eingeben (**CODE-BEISPIEL 1-17**).**

CODE-BEISPIEL 1-17 Diagnoseausgabemeldung `watch-net-all`

```
{0} ok watch-net-all
/pci@1f,0/pci@1,1/network@c,1
Internal loopback test -- succeeded.
Link is -- up
Looking for Ethernet Packets.
'.' is a Good Packet. 'X' is a Bad Packet.
Type any key to stop.
```

Automatische Systemwiederherstellung (Automatic System Recovery, ASR)

Hinweis – Die automatische Systemwiederherstellung (Automatic System Recovery (ASR) ist nicht dasselbe wie der automatische Serverneustart (Automatic Server Restart), der vom Netra 240 Server ebenfalls unterstützt wird. Informationen zum automatischen Serverneustart (Automatic Server Restart) finden Sie in Kapitel 3.

Die automatische Systemwiederherstellung (ASR) besteht aus Selbsttestfunktionen und der Möglichkeit der automatischen Konfiguration, um ausgefallene Hardware-Komponenten zu finden und ihre Konfiguration rückgängig zu machen. Durch Aktivieren von ASR kann der Server den Betrieb nach bestimmten, nicht schwerwiegenden Hardware-Ausfällen oder anderen Fehlern wieder aufnehmen.

Wenn eine Komponente durch ASR überwacht wird und der Server den Betrieb ohne diese Komponente fortsetzen kann, wird der Server automatisch neu gestartet, wenn diese Komponente einen Fehler aufweist oder ausfällt. Dadurch wird verhindert, dass eine defekte Hardwarekomponente das gesamte System außer Betrieb setzt oder ständig abstürzen lässt.

Wenn während des Einschaltens ein Fehler gefunden wird, wird die fehlerhafte Komponente deaktiviert. Wenn das System weiterhin funktionsfähig ist, wird der Startvorgang fortgesetzt.

Zur Unterstützung einer solchen eingeschränkten Startfähigkeit bedient sich die OpenBoot-Firmware der 1275 Client-Schnittstelle (über die Gerätestruktur), um die Geräte als *ausgefallen* oder *deaktiviert* zu kennzeichnen. Zu diesem Zweck wird im Gerätestrukturknoten eine entsprechende Statuseigenschaft erstellt. Das Solaris-Betriebssystem aktiviert für auf diese Weise gekennzeichnete Subsysteme keinen Treiber.

Solange die ausgefallene Komponente elektrisch inaktiv ist (d. h. keine zufälligen Busfehler, Signalrauschen o. ä. verursacht), kann das System auf diese Weise automatisch neu gestartet und wieder in Betrieb genommen werden, während der Kundendienst gerufen wird.

Sobald ein ausgefallenes oder deaktiviertes Gerät durch ein neues ersetzt wird, ändert die OpenBoot-Firmware automatisch den Status des Geräts beim Neustart.

Hinweis – Die ASR-Funktionen müssen durch den Benutzer aktiviert werden (siehe „So aktivieren Sie ASR“ auf Seite 33).

Optionen für automatisches Starten

Die Einstellung `auto-boot?` steuert, ob die Firmware das Betriebssystem automatisch nach jedem Zurücksetzen neu startet. Die Standardeinstellung ist `true`.

Die Einstellung `auto-boot-on-error?` steuert, ob das System versucht, einen eingeschränkten Start durchzuführen, wenn ein Subsystemfehler entdeckt wird. Sowohl die Einstellung `auto-boot?` als auch die Einstellung `auto-boot-on-error?` muss auf `true` gesetzt sein, damit automatisch ein eingeschränkter Startvorgang möglich ist.

- **Mit der folgenden Eingabe nehmen Sie die Einstellung der Schalter vor:**

```
ok setenv auto-boot? true
ok setenv auto-boot-on-error? true
```

Hinweis – Die Standardeinstellung für `auto-boot-on-error?` ist `false`. Das System unternimmt daher erst den Versuch eines eingeschränkten Starts, wenn Sie diese Einstellung in `true` ändern. Außerdem versucht das System auch keinen eingeschränkten Start beim Auftreten von schwerwiegenden, irreparablen Fehlern, auch wenn eigentlich festgelegt wurde, dass eingeschränkte Starts versucht werden sollen. Beispiele für schwerwiegende, irreparable Fehler finden Sie unter „Fehlerbehandlung im Überblick“ auf Seite 32.

Fehlerbehandlung im Überblick

Die Fehlerbehandlung während der Einschaltsequenz kann in drei Kategorien eingeteilt werden:

- Wenn die POST- oder OpenBoot-Diagnoseroutinen keine Fehler entdecken, versucht das System zu starten, sofern `auto-boot?` auf `true` gesetzt ist.
- Wenn die POST- oder/und OpenBoot-Diagnoseroutinen nur nicht schwerwiegende Fehler finden, versucht das System zu starten, sofern für `auto-boot?` die Einstellung `true` und für `auto-boot-on-error?` die Einstellung `true` festgelegt wurde.

Hinweis – Wenn die POST- oder OpenBoot-Diagnoseroutinen einen nicht schwerwiegenden Fehler des normalen Boot-Geräts feststellen, dekonfiguriert die OpenBoot-Firmware automatisch dieses ausgefallene Gerät und versucht, das System über das Boot-Gerät zu starten, das in der mit der Konfigurationsvariable `boot-device` festgelegten Reihenfolge als Nächstes folgt.

- Stellen die POST- oder OpenBoot-Diagnoseroutinen einen schwerwiegenden Fehler fest, wird das System nicht gestartet und zwar unabhängig von den Einstellungen für `auto-boot?` oder `auto-boot-on-error?`. Zu den schwerwiegenden, irreparablen Fehlern gehören:
  - Ausfall aller CPUs
  - Ausfall aller logischen Speicherbänke
  - Ausfall der Flash-RAM-CRC (Cyclical Redundancy Check, zyklische Redundanzprüfung)
  - Ausfall kritischer FRU (Field-Replaceable Unit)-PROM-Konfigurationsdaten
  - Ausfall einer kritischen ASIC (Application Specific Integrated Circuit, anwendungsspezifische integrierte Schaltung)

Szenarien für das Zurücksetzen des Systems

Sie können über drei OpenBoot-Konfigurationsvariablen (`diag-switch?`, `obdiag-trigger` und `post-trigger`) steuern, ob aufgrund eines Systemrücksetzungsereignisses eine Firmware-Diagnoseprüfung durchgeführt wird.

Das Standardrücksetzprotokoll führt die POST- und OpenBoot-Diagnoseroutinen nur dann aus, wenn `diag-switch?` auf `true` gesetzt ist. Die Standardeinstellung für diese Variable ist `false`. Da sich ASR bei der Erkennung von fehlerhaften Geräten auf die Firmware-Diagnoseprüfungen stützt, muss `diag-switch?` auf `true` gesetzt sein, damit ASR ausgeführt wird. Diesbezügliche Anweisungen erhalten Sie unter „So aktivieren Sie ASR" auf Seite 33.

Mithilfe von `obdiag-trigger` und `post-trigger` können Sie steuern, welche Rücksetzereignisse ggf. automatisch Firmware-Diagnoseprüfungen initiieren. Eine ausführliche Erklärung dieser Variablen und ihrer Verwendung finden Sie unter „Steuern der POST-Diagnose" auf Seite 10 sowie unter „Steuerung der OpenBoot-Diagnoseprüfungen" auf Seite 17.

▼ So aktivieren Sie ASR

1. **Geben Sie an der Eingabeaufforderung `ok` die folgende Zeile ein:**

```
ok setenv diag-switch? true
ok setenv auto-boot? true
ok setenv auto-boot-on-error? true
```

2. **Setzen Sie die Variable `obdiag-trigger` auf `power-on-reset`, `error-reset` oder `user-reset`.**

   Geben Sie beispielsweise die folgende Zeile ein:

```
ok setenv obdiag-trigger user-reset
```

3. **Geben Sie die folgende Zeile ein:**

```
ok reset-all
```

Das System speichert die Parameteränderungen dauerhaft und bootet automatisch, wenn die OpenBoot-Variable `auto-boot?` auf `true` (Standardwert) gesetzt ist.

Hinweis – Die Speicherung der Parameteränderungen ist auch durch einen Neustart des Systems über den Netz-/Standbyschalter auf der Vorderseite des Servers möglich.

▼ So deaktivieren Sie ASR

1. **Geben Sie an der Eingabeaufforderung `ok` die folgende Zeile ein:**

```
ok setenv diag-switch? false
```

2. **Geben Sie die folgende Zeile ein:**

```
ok reset-all
```

Das System speichert die Parameteränderung dauerhaft.

Hinweis – Die Speicherung der Parameteränderungen ist auch durch einen Neustart des Systems über den Netz-/Standbyschalter auf der Vorderseite des Servers möglich.

KAPITEL **2**

SunVTS-Software

In diesem Kapitel wird SunVTS beschrieben. In diesem Kapitel werden folgende Themen behandelt:

SunVTS-Software – Überblick

Die SunVTS 5.1 Patch Set 5 (PS5)-Software und zukünftige kompatible Versionen werden vom Netra 240 Server unterstützt.

Die SunVTS-Software, die Sun Validation Test Suite, ist ein Online-Diagnosewerkzeug für das Überprüfen der Konfiguration und der Funktionalität von Hardware-Controllern, Geräten und Plattformen. Es wird unter dem Solaris-Betriebssystem ausgeführt und hat die folgenden Schnittstellen:

- Befehlszeilenschnittstelle (Command-line interface, CLI)
- Serielle Schnittstelle (`ttya`)

Mit der Sun VTS Software Suite werden das System und die Peripheriegeräte Belastungstests unterzogen. Eine Sitzung mit der SunVTS-Software kann über das Netzwerk angezeigt und gesteuert werden. Mithilfe eines Remote-Systems können Sie so eine laufende SunVTS-Testsitzung überwachen, die Testoptionen ändern und alle Testfunktionen für ein anderes System im Netzwerk steuern.

Die SunVTS-Software kann in drei Testmodi laufen:

- *Verbindungsmodus* überprüft das Vorhandensein von Gerätecontrollern. Dieser Test dauert in der Regel nur wenige Minuten und eignet sich gut zur Überprüfung der Systemverbindungen.
- *Funktionaler Modus* untersucht nur die von Ihnen gewählten Untersysteme. Dies ist der Standardmodus.
- *AutoKonfig-Modus* erkennt automatisch alle Untersysteme und prüft sie mit einer der beiden folgenden Methoden:
  - *Zuverlässigkeitstest* führt einen Testdurchgang an allen Untersystemen aus und beendet die Untersuchung danach. Dieser Test dauert bei typischen Systemkonfigurationen ein bis zwei Stunden.
  - *Ausführlicher Test* untersucht sämtliche Untersysteme in mehreren Testdurchläufen bis zu 24 Stunden lang.

Da SunVTS zahlreiche Tests parallel ausführt und dabei beträchtliche Systemressourcen an sich zieht, sollten Sie sich genau überlegen, wann Sie diese Tests in einer Produktionsumgebung durchführen. Während eines Dauertests im ausführlichen Testmodus von SunVTS dürfen auf dem untersuchten System keine anderen Anwendungen laufen.

Auf einem Server muss das Solaris-Betriebssystem ausgeführt werden, damit es von der SunVTS-Software getestet werden kann. Das SunVTS-Softwarepaket ist optional erhältlich, d. h. nicht unbedingt auf Ihrem System vorinstalliert. Diesbezügliche Anweisungen erhalten Sie unter „So stellen Sie fest, ob die SunVTS-Software installiert ist“ auf Seite 38.

SunVTS-Tests

Mit der SunVTS-Software können Sie Testsitzungen auf einem remote angeschlossenen Server anzeigen und steuern. TABELLE 2-1 führt einige der verfügbaren Tests auf:

TABELLE 2-1 SunVTS-Software-Tests

SunVTS-Software-Test	Beschreibung
`cputest`	Prüft die CPU.
`disktest`	Prüft die lokalen Festplattenlaufwerke.
`dvdtest`	Prüft das DVD-ROM-Laufwerk.
`n240atest`	Testet die Alarmkarte für Alarmtransits, LEDs und die FRU ID.
`fputest`	Prüft die Gleitkommaeinheit.

TABELLE 2-1 SunVTS-Software-Tests *(Fortsetzung)*

SunVTS-Software-Test	Beschreibung
`nettest`	Testet die Ethernet-Hardware auf der Systemplatine und die Netzwerk-Hardware auf optionalen PCI-Karten.
`netlbtest`	Führt einen Loopback-Test durch, um zu prüfen, ob der Ethernet-Adapter Pakete senden und empfangen kann.
`pmem`	Prüft den physischen Hauptspeicher (nur Lesen).
`sutest`	Prüft die seriellen Anschlüsse auf der Hauptserverplatine.
`vmem`	Prüft den virtuellen Speicher (eine Kombination aus Swap-Partition und physischem Hauptspeicher).
`env6test`	Testet die Umgebungsgeräte.
`ssptest`	Testet ALOM-Hardware-Geräte.
`i2c2test`	Testet I^2C-Geräte auf korrekten Betrieb.

SunVTS-Software und Sicherheit

Während der Installation von SunVTS müssen Sie sich zwischen den Sicherheitsoptionen Basic oder Sun Enterprise Authentication Mechanism (SEAM) entscheiden. Die Option Basic Security verwendet eine lokale Sicherheitsdatei aus dem Installationsverzeichnis der SunVTS-Software zur Eingrenzung der Benutzer, Gruppen und Hosts, die Zugriff auf die SunVTS-Software haben. Die Option SEAM basiert auf dem Standard-Netzwerkauthentifizierungsprotokoll Kerberos. Sie gewährleistet eine sichere Benutzerauthentifizierung, Datenintegrität und Datenschutz für Transaktionen über Netzwerke.

Wenn Sie an Ihrem Standort die Sicherheitsoption SEAM verwenden, muss die SEAM-Client- und Serversoftware in Ihrer Netzwerkumgebung installiert sein und sowohl in der Solaris- als auch in der SunVTS-Software korrekt konfiguriert sein. Wenn Ihr Standort keine SEAM-Sicherheit verwendet, dann wählen Sie die Option auch nicht während der Installation der SunVTS-Software.

Wenn Sie während der Installation das falsche Sicherheitsschema wählen oder das gewählte Schema falsch konfigurieren, lassen sich die SunVTS-Software-Tests unter Umständen nicht ausführen. Weitere Informationen hierzu finden Sie im *SunVTS User's Guide* und in den Anleitungen zur SEAM-Software.

▼ So stellen Sie fest, ob die SunVTS-Software installiert ist

● **Geben Sie die folgende Zeile ein:**

```
# pkginfo -l SUNWvts
```

- Wenn SunVTS installiert wurde, werden Informationen zum Paket angezeigt.
- Wenn die Software SunVTS nicht installiert wurde, sehen Sie eine Fehlermeldung:

```
ERROR: information for "SUNWvts" was not found
```

Installation der SunVTS-Software

Standardmäßig ist die SunVTS-Software auf dem Netra 240 Server nicht installiert. Sie befindet sich jedoch auf der Supplement-CD, die mit dem Solaris-Betriebssystem ausgeliefert wird, und die aktuellen Versionen können von der folgenden Website heruntergeladen werden:

`http://www.sun.com/oem/products/vts/`

Hinweis – Die SunVTS 5.1 Patch Set 5 (PS5)-Software und zukünftige kompatible Versionen werden vom Netra 240 Server unterstützt.

Weitere Informationen über die SunVTS-Software finden Sie in der SunVTS-Dokumentation, die der Solaris Software-Version entspricht, die Sie ausführen. Zusätzlich finden Sie Informationen zur SunVTS-Software sowie Installationsanweisungen auf der obigen Website.

Anzeigen der Dokumentation für die SunVTS-Software

Die Dokumente der SunVTS-Software befinden sich auf der Software Supplement-CD, die mit jedem Solaris-Media-Kit geliefert wird. Diese Dokumente sind auch verfügbar unter `http://docs.sun.com`.

Weitere Informationen finden Sie in den folgenden SunVTS-Software-Dokumenten:

- Der *SunVTS User's Guide* beschreibt die Installation, die Konfiguration und die Ausführung der SunVTS-Diagnosesoftware.
- Die *SunVTS Quick Reference Card* bietet einen Überblick über die Verwendung der SunVTS-Schnittstelle.
- Das *SunVTS Test Reference Manual* enthält Einzelheiten über jeden einzelnen SunVTS-Test.

KAPITEL **3**

Advanced Lights Out Manager

In diesem Kapitel finden Sie einen Überblick über die Sun™ Advanced Lights Out Manager (ALOM)-Software. Es werden folgende Themen behandelt:

Advanced Lights Out Manager – Überblick

Der Netra 240 Server wird mit installiertem Sun Advanced Lights Out Manager ausgeliefert. Die Systemkonsole wird standardmäßig von ALOM überwacht und ist so konfiguriert, dass Serverkonsolendaten beim Start angezeigt werden.

Mit ALOM können Sie Server entweder über eine serielle Verbindung (über den SERIAL MGT-Anschluss) oder eine Ethernet-Verbindung (über den NET MGT-Anschluss) überwachen und steuern. Informationen über die Konfiguration einer Ethernet-Verbindung finden Sie im *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch* (817-5003-11).

Hinweis – Der serielle ALOM-Anschluss ist mit SERIAL MGT bezeichnet und sollte ausschließlich für die Serververwaltung verwendet werden. Wenn Sie einen seriellen Anschluss für allgemeine Zwecke benötigen, verwenden Sie den seriellen Anschluss 10101.

Sie können ALOM so konfigurieren, dass bei Hardware-Fehlern und anderen Server- oder ALOM-bezogenen Ereignissen E-Mail-Benachrichtigungen gesendet werden.

Die ALOM-Schaltung arbeitet mit einer Standby-Stromversorgung vom Server mit folgenden Ergebnissen:

- ALOM wird aktiv, sobald der Server an eine Stromquelle angeschlossen wird, und bleibt es, bis das Netzkabel abgezogen wird.
- Deshalb bleibt die ALOM-Firmware und -Software auch in Betrieb, wenn das Betriebssystem des Servers vom Netz getrennt wird.

In TABELLE 3-1 werden die Komponenten aufgeführt, die von ALOM überwacht werden, sowie die Informationen, die die Software für jede Komponente ausgibt.

TABELLE 3-1 Von ALOM überwachte Komponenten

Komponente	Zur Verfügung gestellte Informationen
Festplattenlaufwerke	Vorhandensein und Status
System- und CPU-Lüfter	Geschwindigkeit und Status
CPUs	Vorhandensein, Temperatur sowie Warnungen vor Überhitzung und Ausfallmeldungen
Netzteile	Vorhandensein und Status
Systemtemperatur	Umgebungstemperatur sowie Warnungen vor Überhitzung und Ausfallmeldungen
Server-Frontplatte	Drehknopfposition und LED-Status
Spannung	Status und Schwellenwerte
SCSI- und USB-Leistungsschalter	Status
Trockenkontakt-Relais-Alarme	Status

ALOM-Anschlüsse

Der Standardverwaltungsanschluss ist SERIAL MGT. Dieser Anschluss verwendet einen RJ-45-Anschluss und ist *nur* für die Serververwaltung geeignet; er unterstützt nur ASCII-Verbindungen zu einer externen Konsole. Verwenden Sie diesen Anschluss, wenn Sie den Server zum ersten Mal in Betrieb nehmen.

Ein anderer serieller Anschluss – 10101 – ist für die allgemeine serielle Datenübertragung verfügbar. Dieser Anschluss verwendet einen DB-9-Anschluss. Informationen zu Pin-Belegungen finden Sie im *Netra 240 Server Installationshandbuch* (Teilenummer 817-5001-11).

Zusätzlich verfügt der Server über eine 10BASE-T Ethernet-Verwaltungsdomänenschnittstelle mit dem Namen NET MGT. Für die Verwendung dieses Anschlusses ist die ALOM-Konfiguration erforderlich. Informationen hierzu finden Sie im *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch* (Teilenummer 817-5003-11).

Festlegen des Passworts `admin`

Wenn Sie nach dem ersten Einschalten zur ALOM-Software wechseln, wird die Eingabeaufforderung `sc>` angezeigt. Hier können Sie Befehle ausführen, für die keine Benutzerberechtigungen erforderlich sind. (Eine Liste mit Befehlen finden Sie im *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch*, Teilenummer 817-5003-11.) Wenn Sie versuchen, einen Befehl auszuführen, für den eine Benutzerberechtigung erforderlich ist, werden Sie zur Festlegung eines Passworts für den Benutzer `admin` aufgefordert.

- **Richten Sie ein Passwort für den Benutzer `admin` ein, wenn Sie dazu aufgefordert werden.**

Das Passwort muss Folgendes enthalten:

- mindestens zwei Buchstaben
- mindestens eine Ziffer und ein Sonderzeichen
- insgesamt zwischen sechs und acht Zeichen

Sobald das Passwort festgelegt ist, verfügt der Benutzer `admin` über die vollständige Berechtigung und kann alle ALOM CLI-Befehle ausführen. Der Benutzer wird zur Anmeldung mit dem admin-Passwort aufgefordert, wenn er anschließend zu ALOM wechselt.

ALOM-Grundfunktionen

In diesem Abschnitt werden einige ALOM-Grundfunktionen beschrieben. Die umfassende Dokumentation finden Sie im *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch* (Teilenummer 817-5003-11) und den *Netra 240 Server Versionshinweisen* (817-3142-xx).

▼ So wechseln Sie zur ALOM-Eingabeaufforderung

- **Geben Sie an der Befehlseingabeaufforderung die folgende `#.`-Tastenfolge ein:**

```
# #.
```

Hinweis – Wenn Sie zur ALOM-Eingabeaufforderung wechseln, sind Sie mit der Benutzer-ID `admin` angemeldet. Siehe „Festlegen des Passworts `admin`“ auf Seite 43.

▼ So wechseln Sie zur Serverkonsolen-Eingabeaufforderung

- **Geben Sie die folgende Zeile ein:**

```
sc> console
```

Es können zwar mehrere ALOM-Benutzer gleichzeitig mit dem Server verbunden sein, aber nur ein Benutzer kann Zeichen an der Konsole eingeben.

Wenn ein anderer Benutzer angemeldet ist und über Schreibrechte verfügt, wird die folgende Meldung nach Eingabe des Befehls `console` angezeigt:

```
sc> Console session already in use. [view mode]
```

▼ So entziehen Sie einem anderen Benutzer die Schreibrechte

- **Geben Sie die folgende Zeile ein:**

```
sc> console -f
```

Automatischer Serverneustart

Hinweis – Die automatische Systemwiederherstellung (Automatic System Recovery, ASR) ist nicht dasselbe wie der automatische Serverneustart (Automatic Server Restart), der vom Netra 240 Server ebenfalls unterstützt wird.

Der automatische Serverneustart ist eine Komponente von ALOM. Er überwacht das Solaris-Betriebssystem während dessen Ausführung, synchronisiert standardmäßig die Dateisysteme und startet den Server bei einem Ausfall neu.

ALOM verwendet einen Watchdog-Vorgang, um *nur* den Kernel zu überwachen. ALOM startet den Server nicht neu, wenn ein Vorgang hängt und der Kernel noch ausgeführt wird. Die ALOM-Watchdog-Parameter für das Watchdog-Patting-Intervall und die Watchdog-Zeitüberschreitung können nicht vom Benutzer konfiguriert werden.

Wenn der Kernel hängt und eine Watchdog-Zeitüberschreitung eintritt, berichtet und protokolliert ALOM das Ereignis und führt eine von drei Aktionen durch, die vom Benutzer konfiguriert werden können:

- xir – Durch diese Standardaktion wird der Server zum Synchronisieren der Dateisysteme und zum Neustart aufgefordert. Wenn die Synchronisierung hängt, führt ALOM nach 15 Minuten einen Kaltstart durch.
- Reset – Dies ist ein Kaltstart, der zu einer schnellen Systemwiederherstellung führt. Es werden allerdings keine Diagnosedaten über das Hängen gespeichert und es kann zu schwerwiegenden Schäden kommen.
- None – Das System wird im hängenden Zustand belassen, nachdem die Watchdog-Zeitüberschreitung berichtet wurde.

Weitere Informationen finden Sie im Abschnitt `sys_autorestart` des *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuchs* (Teilenummer 817-5003-11).

Anweisungen über die Verwendung der automatischen Systemwiederherstellung (ASR) finden Sie in Kapitel 1.

Überwachen und Steuern der Umgebungsbedingungen

Der Netra 240 Server verfügt über ein Subsystem zur Überwachung der Umgebungsbedingungen, das den Server und seine Komponenten vor Folgendem schützen soll:

- extreme Temperaturen
- kein adäquater Luftstrom im System
- Betrieb mit fehlenden oder falsch konfigurierten Komponenten
- Ausfall der Stromversorgung
- interne Hardwarefehler

Die Überwachungs- und Steuerungsfunktionen werden von der ALOM-Firmware verwaltet, mit der sichergestellt wird, dass die Überwachungsfunktionen auch dann aktiv bleiben, wenn das System angehalten wurde oder nicht gestartet werden kann. Zusätzlich können durch die Überwachung mit der ALOM-Firmware Kapazitäten auf dem System freigegeben werden, um CPU- und Speicherressourcen für das Betriebssystem und die Anwendungssoftware freizugeben.

Für das Subsystem zur Überwachung der Umgebungsbedingungen kommt ein I^2C-Bus nach Industriestandard zum Einsatz. Der I^2C-Bus ist ein einfacher, aus zwei Drähten bestehender serieller Bus, der das gesamte System durchzieht, um die Überwachung und Steuerung der Temperatursensoren, der Lüfter, der Netzteile, der Status-LEDs und des Drehschalters für die Systemsteuerung auf der Vorderseite des Systems zu ermöglichen.

Der Server umfasst drei Temperatursensoren, die die Umgebungstemperatur des Servers und die Temperatur der beiden CPUs überwachen. Das Überwachungssubsystem ruft die Werte der einzelnen Sensoren ab, meldet anhand der so gemessenen Temperaturen alle Überhitzungs- oder Unterkühlungszustände und leitet gegebenenfalls entsprechende Maßnahmen ein. Weitere I^2C-Geräte stellen das Fehlen von Komponenten und Komponentenfehler fest.

Durch das Zusammenspiel von Hardware und Software wird sichergestellt, dass die Temperaturen innerhalb des Gehäuses nicht die vordefinierten Bereiche für einen sicheren Betrieb unter- oder überschreiten. Wenn die Temperatur, die durch einen Sensor überwacht wird, die Unterkühlungswarnschwelle unter- oder die Überhitzungswarnschwelle überschreitet, aktiviert die Software des Überwachungssubsystems die LEDs „Wartung erforderlich" auf der Vorder- bzw. Rückseite des Systems. Falls der Überhitzungs- bzw. Unterkühlungszustand andauert und einen „weichen" Schwellenwert bei zu hoher oder zu niedriger Temperatur erreicht, wird das System ordnungsgemäß ausgeschaltet. Falls der Überhitzungs- bzw. Unterkühlungszustand andauert und einen „harten" Schwellenwert bei zu hoher oder zu niedriger Temperatur erreicht, wird das Ausschalten des Systems erzwungen.

Fehler- und Warnmeldungen werden an die Systemkonsole gesendet und in der Datei `/var/adm/messages` protokolliert. Die Wartungs-LEDs leuchten nach einem automatischen Herunterfahren des Systems weiter, um die Problemdiagnose zu unterstützen.

Die Arten der Meldungen, die an die Systemkonsole gesendet und in der Datei `/var/adm/messages` protokolliert werden, hängen von Ihrer Festlegung der ALOM-Benutzervariablen `sc_clieventlevel` und `sys_eventlevel` ab. Informationen zum Festlegen dieser Variablen finden Sie im *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch* (817-5003-11).

TABELLE 3-2 Temperaturschwellenwerte für das Netra 240 Server-Gehäuse

Temperaturschwellenwert	Temperatur	Serveraktion
Unterkühlung, „hartes“ Herunterfahren	-11°C	Server initiiert das erzwungene Herunterfahren des Systems.
Unterkühlung, „weiches“ Herunterfahren	-9°C	Server initiiert das ordnungsgemäße Herunterfahren des Systems.
Unterkühlungswarnung	-7°C	Die Systemwartungs-LEDs an der Vorder- und Rückseite des Servers leuchten auf.
Überhitzungswarnung	57°C	Die Systemwartungs-LEDs an der Vorder- und Rückseite des Servers leuchten auf.
Überhitzung, „weiches“ Herunterfahren	60°C	Server initiiert das ordnungsgemäße Herunterfahren des Systems.
Überhitzung, „hartes“ Herunterfahren	63°C	Server initiiert das erzwungene Herunterfahren des Systems.

Das Überwachungssubsystem wurde so entwickelt, dass es auch eventuelle Ausfälle der vier Systemgebläse erkennt. Wenn irgendein Gebläse ausfällt, erkennt das Überwachungssubsystem den Ausfall, sendet eine entsprechende Fehlermeldung an die Systemkonsole, protokolliert diese Meldung in der Datei `/var/adm/messages` und aktiviert die Wartungs-LEDs.

Das Stromversorgungssubsystem wird auf ähnliche Weise überwacht. Das Überwachungssubsystem fragt von Zeit zu Zeit die Statusregister des Netzteils ab und zeigt den Status der Leistungsabgabe, des Leistungseingangs und das Vorhandensein eines jeden Netzteils an.

Erkennt das Subsystem ein Problem mit einem Netzteil, wird eine entsprechende Fehlermeldung an die Systemkonsole gesendet und in der Datei `/var/adm/messages` protokolliert. Darüber hinaus leuchten LEDs an jedem Netzteil auf, um auf den Fehler aufmerksam zu machen. Die Wartungs-LED leuchtet, um auf einen Systemfehler hinzuweisen. Die ALOM-Konsolenmeldungen verzeichnen Ausfälle der Netzteile.

Verwenden Sie den ALOM-Befehl `showenvironment`, um die Warnschwellenwerte des Stromversorgungssubsystems und die Lüftungsgeschwindigkeiten anzuzeigen. Informationen zur Verwendung dieses Befehls finden Sie im *Sun Netra 240 Server Advanced Lights Out Manager-Benutzerhandbuch* (Teilenummer 817-5003-11).

ANHANG **A**

Alarm-Relay-Ausgaben-Anwendungsprogrammier-schnittstelle

Dieser Anhang stellt ein Beispielprogramm zur Verfügung (CODE-BEISPIEL A-1), das verdeutlicht, wie mit den Befehlen `get/set` der Status des Alarms abgerufen bzw. festgelegt wird. Die Anwendung kann `LOMIOCALSTATE` ioctl verwenden, um den Status eines jeden Alarms abzurufen, sowie `LOMIOCALCTL` ioctl, um die Alarme individuell festzulegen. Weitere Einzelheiten über die Alarmanzeigen finden Sie im *Netra 240 Server Service Manual* (817-2699).

CODE-BEISPIEL A-1 Beispielprogramm für den `get/set`-Status des Alarms

```
#include <sys/types.h>
#include <string.h>
#include <stdlib.h>
#include <sys/unistd.h>
#include <fcntl.h>
#include "lom_io.h"

#define ALARM_INVALID   -1
#define LOM_DEVICE  "/dev/lom"

static void usage();
static void get_alarm(const char *alarm);
static int set_alarm(const char *alarm, const char *alarmval);
static int parse_alarm(const char *alarm);
static int lom_ioctl(int ioc, char *buf);
static char *get_alarmval(int state);
static void get_alarmvals();


main(int argc, char *argv[])
{
```

CODE-BEISPIEL A-1 Beispielprogramm für den `get/set`-Status des Alarms *(Fortsetzung)*

```
#include <sys/types.h>

        if (argc < 3) {
                usage();
                if (argc == 1)
                        get_alarmvals();
                exit (1);
        }

        if (strcmp(argv[1], "get") == 0) {
                if (argc != 3) {
                        usage();
                        exit (1);
                }
                        get_alarm(argv[2]);
        }
        else
        if (strcmp(argv[1], "set") == 0) {
                if (argc != 4) {
                        usage();
                        exit (1);
                }
                set_alarm(argv[2], argv[3]);
        } else {
                usage();
                exit (1);
        }
}

static void
usage()
{
        printf("usage: alarm [get|set] [crit|major|minor|user] [on|off]\n");
}

static void
get_alarm(const char *alarm)
{
        ts_aldata_t     ald;
        int altype = parse_alarm(alarm);
        char *val;

        if (altype  == ALARM_INVALID) {
                usage();
                exit (1);
        }
```

CODE-BEISPIEL A-1 Beispielprogramm für den `get/set`-Status des Alarms *(Fortsetzung)*

```
#include <sys/types.h>
        ald.alarm_no = altype;
        ald.alarm_state = ALARM_OFF;

        lom_ioctl(LOMIOCALSTATE, (char *)&ald);

        if ((ald.alarm_state != ALARM_OFF) &&
                        (ald.alarm_state != ALARM_ON)) {
                printf("Invalid value returned: %d\n", ald.alarm_state);
                exit (1);
        }

        printf("ALARM.%s = %s\n", alarm, get_alarmval(ald.alarm_state));
}

static int
set_alarm(const char *alarm, const char *alarmstate)
{
        ts_aldata_t     ald;
        int alarmval = ALARM_OFF, altype = parse_alarm(alarm);

        if (altype  == ALARM_INVALID) {
                usage();
                exit (1);
        }

        if (strcmp(alarmstate, "on") == 0)
                alarmval = ALARM_ON;
        else
        if (strcmp(alarmstate, "off") == 0)
                alarmval = ALARM_OFF;
        else {
                usage();
                exit (1);
        }

        ald.alarm_no = altype;
        ald.alarm_state = alarmval;

        if (lom_ioctl(LOMIOCALCTL, (char *)&ald) != 0) {
                printf("Setting ALARM.%s to %s failed\n", alarm, alarmstate);
                return (1);
        } else {
                printf("Setting ALARM.%s successfully set to %s\n", alarm,
alarmstate);
                return (1);
        }
```

CODE-BEISPIEL A-1 Beispielprogramm für den `get/set`-Status des Alarms *(Fortsetzung)*

```
#include <sys/types.h>
}

static int
parse_alarm(const char *alarm)
{
        int altype;

        if (strcmp(alarm, "crit") == 0)
                altype = ALARM_CRITICAL;
        else
        if (strcmp(alarm, "major") == 0)
                altype = ALARM_MAJOR;
        else
        if (strcmp(alarm, "minor") == 0)
                altype = ALARM_MINOR;
        else
        if (strcmp(alarm, "user") == 0)
                altype = ALARM_USER;
        else {
                printf("invalid alarm value: %s\n", alarm);
                altype = ALARM_INVALID;
        }

        return (altype);

}

static int
lom_ioctl(int ioc, char *buf)
{
        int fd, ret;

        fd = open(LOM_DEVICE, O_RDWR);

        if (fd == -1) {
                printf("Error opening device: %s\n", LOM_DEVICE);
                exit (1);
        }

        ret = ioctl(fd, ioc, (void *)buf);

        close (fd);

        return (ret);
}
```

CODE-BEISPIEL A-1 Beispielprogramm für den `get/set`-Status des Alarms *(Fortsetzung)*

```
#include <sys/types.h>
static char *
get_alarmval(int state)
{
        if (state == ALARM_OFF)
                return ("off");
        else
        if (state == ALARM_ON)
                return ("on");
        else
                return (NULL);
}
static void
get_alarmvals()
{
        get_alarm("crit");
        get_alarm("major");
        get_alarm("minor");
        get_alarm("user");

}
```

Index